Informatique

TP n°3 - Numpy

Application à la représentation graphique et à la résolution d'équations différentielles

PSI - Lycée Rabelais

1 Numpy et représentation graphique

On s'intéresse à l'étude de la fonction $f(t) = \dfrac{e^t}{t+1}$ où t est un réel positif. La bibliothèque `numpy` est importée par l'instruction `import numpy as np`.

Question 1. Définir une fonction f qui prend en argument t et renvoie $f(t)$.

Question 2. Créer un vecteur `numpy`, noté `V_t`, comprenant n valeurs allant de 0 à T.

Question 3. Tracer la représentation de la fonction f sur l'intervalle $[0, T]$ en important le sous-module `pyplot` de la bibliothèque `matplotlib`.

Question 4. Reprendre les questions précédentes en utilisant des listes. On nommera `liste_t` la liste contenant les valeurs de t, et `liste_f` la liste contenant $f(t)$. Commenter.

Question 5. On cherche à calculer la composée de f par elle-même, i.e. $f(f(t))$. Écrire les instructions nécessaires pour représenter graphiquement cette nouvelle fonction. On utilisera d'abord un tableau `numpy`, puis des listes.

2 Équations différentielles d'ordre 1

2.1 Premier problème

On considère l'équation différentielle $y' = 4^{-y} + 1$ avec la condition initiale $y(0) = 0$ sur l'intervalle $[0;2]$.

Compléter le programme suivant afin qu'il construise la solution approchée de cette équation différentielle par le schéma d'Euler avec un pas de 0.01 puis qu'il trace cette solution.

```python
import matplotlib.pyplot as plt
import numpy as np
h = 0.01
Tf = 2.
n = ......
Y0 = 0.
Y = [Y0]*...
for i in range(.... , ....):
  Y[i+1] = ......

X = [i*h for i in range(.... , ....)]
plt.plot(.... , ....)
```

2.2 Deuxième problème

On considère maintenant l'équation différentielle $y'(t) + 2y(t) = \frac{1}{2}\sin(10.t)$ avec la condition initiale $y(0) = 2$ sur l'intervalle $[0; 10]$.

Mettre en place une procédure permettant de tracer une solution approchée de cette équation différentielle en imposant n le nombre de subdivisions.

Tester avec $n \in \{10, 50, 100, 1000, 10000\}$. **Conclure**.

3 Balistique

En 1685 François Blondel, ingénieur du Roi, est le premier à décrire la "bonne trajectoire" d'un boulet de canon dans son livre « *Art de jeter les bombes* ».

La modélisation prend notamment en compte les frottement visqueux de l'air sur le boulet de canon. Ces frottement sont supposés proportionnels à la vitesse du boulet. Après modélisation, on peut se ramener au système d'équations différentielles suivantes :

$$\begin{cases} m\ddot{x} = -h\dot{x} \\ m\ddot{y} = -h\dot{y} - mg \end{cases}$$

On prendra les valeurs numériques suivantes : $m = 5.8$ kg ; $h = 0.5$ N/(m/s). Les conditions initiales sont : $x(0) = 0.$ m, $y(0) = 0.$ m, $\dot{x}(0) = v_0.\cos\alpha$ et $\dot{y}(0) = v_0.\sin\alpha$ avec $v_0 = 450$ m/s et α, l'inclinaison du canon que l'on fera varier entre 10° et 90°.

Question 1. Transformer le système d'équations différentielles proposé en une équation différentielle vectorielle d'ordre 1. On posera $Z(t) = \begin{pmatrix} x(t) \\ y(t) \\ \dot{x}(t) \\ \dot{y}(t) \end{pmatrix}$. Mettre cette équation sous la forme $\dot{Z}(t) = F(Z(t))$.

Question 2. Écrire une fonction `F(Z)` qui renvoie une liste contenant les valeurs approchées de $F(Z)$ dans l'ordre spécifié ci-dessus.

Question 3. Compléter le code ci-dessous afin de tracer la trajectoire du boulet dans le plan xy. On utilise ici une boucle `while` afin d'arrêter le calcul lorsque le boulet touche le sol, c'est-à-dire lorsque $y = 0$.

```
import matplotlib.pyplot as plt
import numpy as np
# Declaration des variables globales
h = 0.1
v0 = 450 #m/s
m = 5.8 #kg
h = 0.5 #N/(m/s)
g = 9.81 #m/s^2
## Definition de F(Z)
def F(Z):
    zprim0 = ......
    zprim1 = ......
    zprim2 = ......
    zprim3 = ......
```

```
    return ..................................

def balistique(alpha): ## alpha en degres
    alpha = alpha*np.pi/180. # conversion en radians

    ## Conditions initiales et preparation stockage
    Z0 = ..................................
    T0 = 0.
    Z = [.....]
    T = [.....]
    i = 0
    while .......... :
        t_suivant = .....
        FZi = .....
        x_suivant = ...............
        y_suivant = ...............
        xprim_suivant = ...............
        yprim_suivant = ...............
        Z_suivant = ..............................
        i += 1
        Z.append(..........)
        T.append(..........)

    Z.pop() ## pour enlever le dernier point ou y<0
    T.pop()
    return [T,Z]

## definition de la liste des alpha
liste_alpha = [.......... for k in range(..........)]
for alpha in liste_alpha:
    res = balistique(alpha)
    tps = .......... # recuperation de la liste des temps
    Z = .......... # recuperation de la liste des listes Z

    x = [Z[k][0] for k in range(len(Z))] # recuperation des xk de la liste des listes Z
    y = [Z[k][1] for k in range(len(Z))] # recuperation des yk de la liste des listes Z

    plt.plot(.......) ## trace des yk en fonction des xk
    plt.axis('equal') ## pour forcer la meme echelle sur x et y
    plt.grid()
```

Questions supplémentaires optionnelles :

- Compléter votre script pour déterminer la distance maximale atteinte par le boulet pour un angle donné puis déterminer l'angle optimal pour atteindre une position lointaine.

- On considère une cible carrée qui est le domaine compris dans le carré dont les sommets sont A, B, C et D de coordonnées respectifs : $(xcmin, ycmin)$, $(xcmin, ycmax)$, $(xcmax, ycmax)$, $(xcmax, ycmin)$ où $xcmin = ycmin = 1900$ m et $xcmax = ycmax = 2100$ m. Écrire une fonction `tapedanscible(alpha)` qui revoie `True` si le tir a tapé dans la cible et `False` sinon.
- Écrire une procédure permettant de tracer tous les angles (arrondis au degré près) qui ont tapé dans la cible. On pourra également représenter la cible.

4 Méthode de Runge-Kutta

De nombreuses autres méthodes sont basées sur un principe similaire à celle d'Euler, avec une meilleurs efficacité. On étudiera ici, pour exemple, la méthode de Runge-Kutta. Avec les mêmes notations que dans le cours, y_{k+1} est ici calculé à partir de y_k par :

$$y_{k+1} = y_k + (d_1 + 2d_2 + 2d_3 + d_4)\frac{h}{6}$$

où : $d_1 = F(t_k; y_k)$, $d_2 = F(t_k + \frac{h}{2}; y_k + \frac{h}{2}d_1)$, $d_3 = F(t_k + \frac{h}{2}; y_k + \frac{h}{2}d_2)$ et $d_4 = F(t_k + h; y_k + hd_3)$.

On montre que la méthode de Runge-Kutta est d'ordre 4 ; et donc la solution approchée obtenue est plus précise qu'avec la méthode d'Euler (qui est d'ordre 1). Par contre, le nombre d'opérations à effectuer augmente (tout en restant de complexité linéaire).

Travail demandé : Reprendre le problème 2.2 mais utiliser cette fois-ci la méthode de Runge-Kutta (utiliser impérativement une fonction intermédiaire `F(t,Y)`). On pourra faire le tracé sur la même figure afin de comparer la précision.

5 Mouvement d'une planète

On considère une planète de masse m soumise à l'attraction gravitationnelle d'une étoile de masse M. On fixe un repère centré sur l'étoile $(O, \vec{u}_x, \vec{u}_y)$ dans lequel la planète est repérée par ses coordonnées $(x(t), y(t))$. La projection de la seconde loi de NEWTON dans ce repère permet d'écrire le système d'équation suivante :

$$\begin{cases} \dfrac{\mathrm{d}^2x}{\mathrm{d}t^2}(t) = -GM\dfrac{x(t)}{(x(t)^2 + y(t)^2)^{3/2}} \\ \dfrac{\mathrm{d}^2y}{\mathrm{d}t^2}(t) = -GM\dfrac{y(t)}{(x(t)^2 + y(t)^2)^{3/2}} \end{cases}$$

avec G la constante gravitationnelle. Pour simplifier les calculs numériques, on considèrera que $GM = 1$.

Question 1. Transformer le système d'équations différentielles proposé en une équation différentielle du premier ordre que l'on écrira sous la forme $Y'(t) = F(t, Y(t))$. On définira les vecteurs $Y(t)$ en fonction de $x(t)$, $y(t)$ et de leurs dérivées premières. On explicitera de même $F(t, Y(t))$.

Question 2. En vous inspirant des exercices précédents et des exemples du cours, écrire un ensemble d'instructions permettant de résoudre ce système d'équations différentielles. On prendra pour valeurs valeurs numériques $t_0 = 0.$, $T = 10.$, $x(0) = 1.0$, $y(0) = 0.$, $x'(0) = 0.$ et $y'(0) = 0.7$. Le nombre de divisions de l'intervalle temporel sera fixé supérieur à 100000.

Question 3. Représentez la trajectoire $y(x)$ de la planète sous ces conditions.